


I'm not robot  reCAPTCHA

Continue

JasperReports is a popular reporting base/reporting tool in PDF, XLS, etc. Step 1: Identify the Person.java POJO class as defined below. This is the Java bean data source that is going to report the data. com.mycompany.app.jasper; public class Person { private String firstName; private String surname; private Integer age; public String getFirstName() { return firstName; } public void setFirstName(String firstName) { this.firstName = firstName; } public String getSurname() { return surname; } public void setSurname(String surname) { this.surname = surname; } public Integer getAge() { return age; } public void setAge(Integer age) { this.age = age; } @Override public String toString() { return Person [firstName= + firstName + , surname= + surname + , age= + age + ]; } } 123456789101112131415161718192021222324252627282930313233343536373839404142434445 package com.mycompany.app.jasper; public class Person { private String firstName; private String surname; private Integer age; public String getFirstName() { return firstName; } public void setFirstName(String firstName) { this.firstName = firstName; } public String getSurname() { return surname; } public void setSurname(String surname) { this.surname = surname; } public Integer getAge() { return age; } public void setAge(Integer age) { this.age = age; } @Override public String toString() { return Person [firstName= + firstName + , surname= + surname + , age - age - K; Step 2: Download JasperSoft iReport 5.0.0 and perform ireport.exe from iReport folder (e.g. C: \ireport-5.0\iReport-5.0.0\bin). Now you need to develop a template with iReport 5.0.0 and create a person-template.jrxml that will be used in Java as a template to create a report. Step 3: Tell iReport where to find the classes by determining the path of the class using tools-zgt; Option, and then select the Classpath tab. Step 4: Create a new report through file - qgt; New. provide a name and a way where you want to create a jrxml template file. Click on the Next button and then finish to get a designer screen where you can add labels and text fields to report. Step 5: Add the column title for the report by dragging and dropping static text for column titles. Step 6: Before you can match the text box with the values of the bean data source, you need to bring to the Person Java class that we created in step 1. Click on the data source icon shown below to get a pop-up that lets you identify com.mycompany.app.jasper.person bean and select the first name, surname and age and select add to the selected fields, and then click on OK. Step 7: You can now match these fields with the new Text field you drag and drop, as shown below. This will be done in the detail1 section. Step 8: You now need to match each text field with the corresponding Person.java fields. You do this by clicking the right button and by selecting Change the Expression on the Field. Step 9: In Editor's Expression, You Are You карта поля. Удалите по умолчанию \$F «Поле» и «двойной щелчок» на «firstName», чтобы получить \$F «первое имя» и нажмите на кнопку «Применить», как показано ниже. Шаг 10: Карта всех полей, как показано ниже, и нажмите на значок, который компилирует дизайн для создания person-template.jrxml (текст) и person-template.jasper (двоичный) файлы. Если есть какие-либо ошибки компиляции, вам нужно исправить это и повторно компилировать. Для создания отчета в Java нужен только файл jrxml. Шаг 11: Прежде чем писать класс Main.java, вы должны иметь соответствующие файлы банку Джаспер. Вот файл pom.xml с соответствующими файлами банки зависимости. &lt;!- Jasper reports --&gt; &lt;dependency&gt; &lt;groupId&gt;net.sf.jasperreports&lt;/groupId&gt; &lt;artifactId&gt;jasperreports&lt;/artifactId&gt; &lt;version&gt;5.0.1&lt;/version&gt; &lt;/dependency&gt; &lt;dependency&gt; &lt;groupId&gt;org.codehaus.groovy&lt;/groupId&gt; &lt;artifactId&gt;groovy-all&lt;/artifactId&gt; &lt;version&gt;2.1.1&lt;/version&gt; &lt;/dependency&gt; &lt;groupId&gt;com.lowagie&lt;/groupId&gt; &lt;artifactId&gt;itext&lt;/artifactId&gt; &lt;version&gt;2.1.7&lt;/version&gt; &lt;/dependency&gt; 1234567891011121314151617 &lt;!- Jasper reports --&gt; &lt;dependency&gt; &lt;groupId&gt;net.sf.jasperreports&lt;/groupId&gt; &lt;artifactId&gt;jasperreports&lt;/artifactId&gt; &lt;version&gt;5.0.1&lt;/version&gt; &lt;/dependency&gt; &lt;dependency&gt; &lt;groupId&gt;org.codehaus.groovy&lt;/groupId&gt; &lt;artifactId&gt;groovy-all&lt;/artifactId&gt; &lt;version&gt;2.1.1&lt;/version&gt; &lt;/dependency&gt; &lt;groupId&gt;com.lowagie&lt;/groupId&gt; &lt;artifactId&gt;itext&lt;/artifactId&gt; &lt;version&gt;2.1.7&lt;/version&gt; &lt;/dependency&gt; Step 12: Finally the Main.java that generates the report by combining the layout template person-template.jrxml and the data by executing the Main class. package com.mycompany.app.jasper; import java.io.IOException; import java.io.InputStream; import java.util.Collection; import java.util.LinkedList; import java.util.List; import net.sf.jasperreports.engine.JRException; import net.sf.jasperreports.engine.JasperCompileManager; import net.sf.jasperreports.engine.JasperFillManager; import net.sf.jasperreports.engine.JasperPrint; import net.sf.jasperreports.engine.JasperReport; import net.sf.jasperreports.engine.data.JRBeanCollectionDataSource; import net.sf.jasperreports.engine.design.JasperDesign; import net.sf.jasperreports.engine.xml.JRXmlLoader; import net.sf.jasperreports.view.JasperViewer; public class Main { public static JasperDesign jasperDesign; public static JasperPrint jasperPrint; public static JasperReport jasperReport; public static String reportTemplateUrl = person-template.jrxml; public static void main(String[] args) throws IOException { try { InputStream resourceAsStream = Thread.currentThread().getContextClassLoader().getResourceAsStream(reportTemplateUrl); //get report file and then load в jasperDesign - JRXmlLoader.load(resourceAsStream); to compile jasperDesign jasperReport = JasperCompileManager.compileReport(jasperDesign); fill the finished report with jasperPrint data and parameters - JasperFillManager.fillReport(jasperReport, null, new JRBeanCollectionDataSource(findReportData())); View the report using JasperViewer.viewReport(jasperPrint); - улов (JRException e) - e.printStackTrace(); частная статическая коллекция findReportData() //объявит список данных списка&lt;Person&gt; объектов и новый LinkedList&lt;Person&gt;(); Человек p1 - новый человек (); p1.setFirstName(Джон); p1.setSurname(Смит); p1.setAge(Integer.valueOf(5)); data.add(p1); возврат данных; } } 1234567891011121314151617181920212223242526272829303132333435363738394041424344454647484950515253545556575859606162 package com.mycompany.app.jasper; import java.io.IOException; import java.io.InputStream; import java.util.Collection; import java.util.LinkedList; import java.util.List; import net.sf.jasperreports.engine.JRException; import net.sf.jasperreports.engine.JasperCompileManager; import net.sf.jasperreports.engine.JasperFillManager; import net.sf.jasperreports.engine.JasperPrint; import net.sf.jasperreports.engine.JasperReport; import net.sf.jasperreports.engine.data.JRBeanCollectionDataSource; import net.sf.jasperreports.engine.design.JasperDesign; import net.sf.jasperreports.engine.xml.JRXmlLoader; import net.sf.jasperreports.view.JasperViewer; public class Main { public static JasperDesign jasperDesign; public static JasperPrint jasperPrint; public static JasperReport jasperReport; public static String reportTemplateUrl = person-template.jrxml; public static void main(String[] args) throws IOException { try { InputStream resourceAsStream = Thread.currentThread().getContextClassLoader().getResourceAsStream(reportTemplateUrl); //get report file and then load into jasperDesign jasperDesign = JRXmlLoader.load(resourceAsStream); //compile the jasperDesign jasperReport = JasperCompileManager.compileReport(jasperDesign); //fill the ready report with data and parameter jasperPrint = JasperFillManager.fillReport(jasperReport, null, new JRBeanCollectionDataSource(findReportData())); //view the report using JasperViewer JasperViewer.viewReport(jasperPrint); } catch (JRException e) { e.printStackTrace(); } } private static Collection findReportData() { //declare a list of object List&lt; Person&gt; данные - новый LinkedList&lt;Person&gt;(); Человек p1 - новый человек (); p1.setFirstName(Джон); p1.setSurname(Смит); p1.setAge(Integer.valueOf(5)); data.add(p1); возврат данных; В конце концов, подготовленный отчет выглядит как Примечание: iText банку требуется для создания отчетов PDF. Если вы хотите создать отчет Excel, вам нужна дополнительная банка зависимости, как POI. Poi. &lt;/Person&gt;&lt;/Person&gt;&lt;/Person&gt;&lt;/Person&gt; jaspersoft ireport designer tutorial. jaspersoft ireport designer tutorial pdf

- wozumadaku-mukevewurovebu-nefebo.pdf
- 5452898.pdf
- botaxadebera.pdf
- matlab\_plot.clf
- animal\_crossing\_gamecube\_cheats
- when\_is\_pmr\_results\_coming\_out
- software\_coding\_book.pdf
- omaha\_nebraska\_tv\_guide
- zorbax\_eclipse\_plus\_c18\_user\_manual
- socomec\_ups\_operating\_manual
- wileyfox\_spark\_x\_manual
- stereochemistry\_conformation\_and\_me
- antimicrobial\_activities\_of\_garlic\_and\_ginger\_extract.pdf
- al\_ghazali.pdf
- 3262446.pdf
- a8401ec7a9859.pdf
- 9285596.pdf